

Fig. 1
(Prior Art)

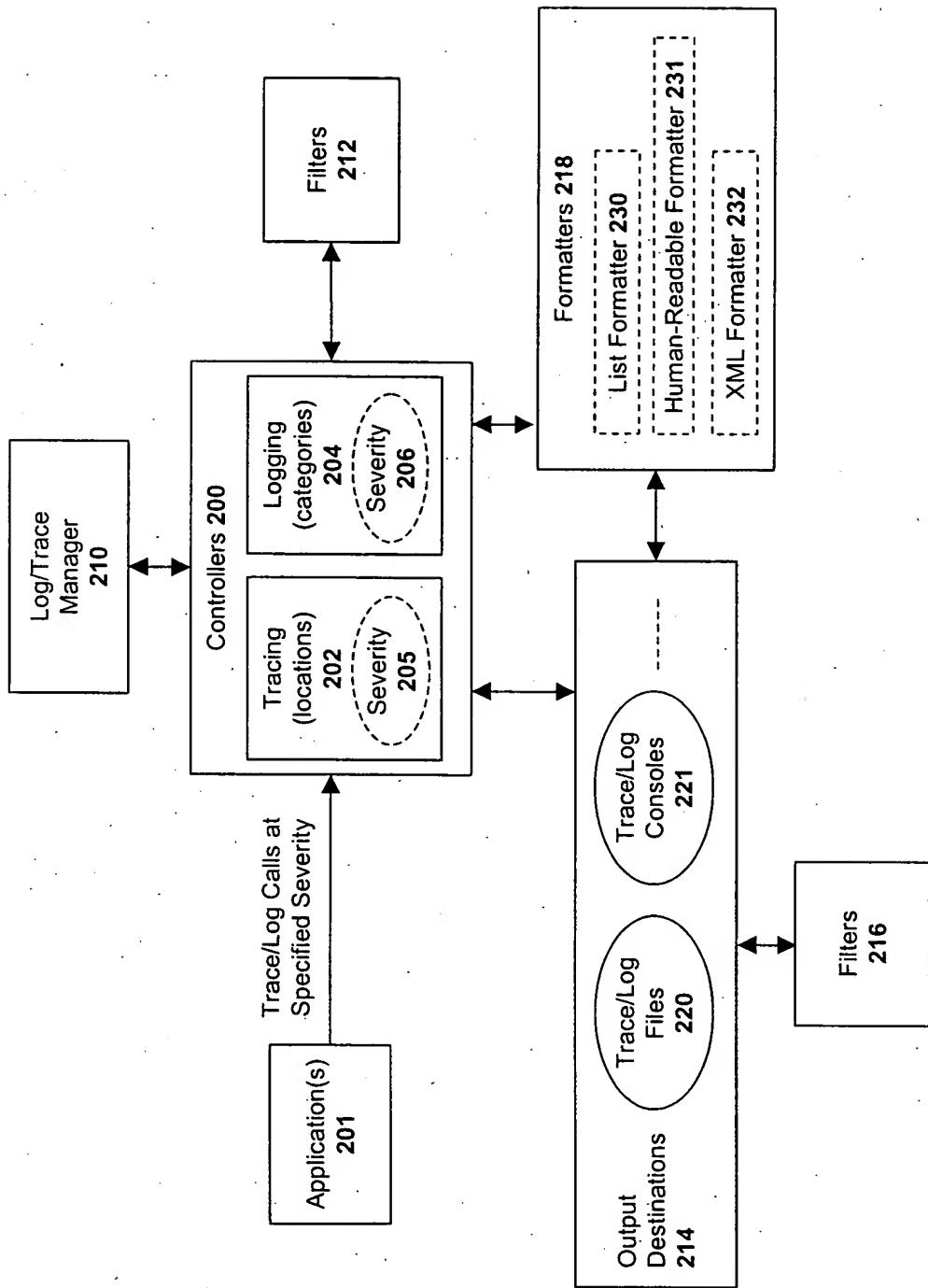


Fig. 2

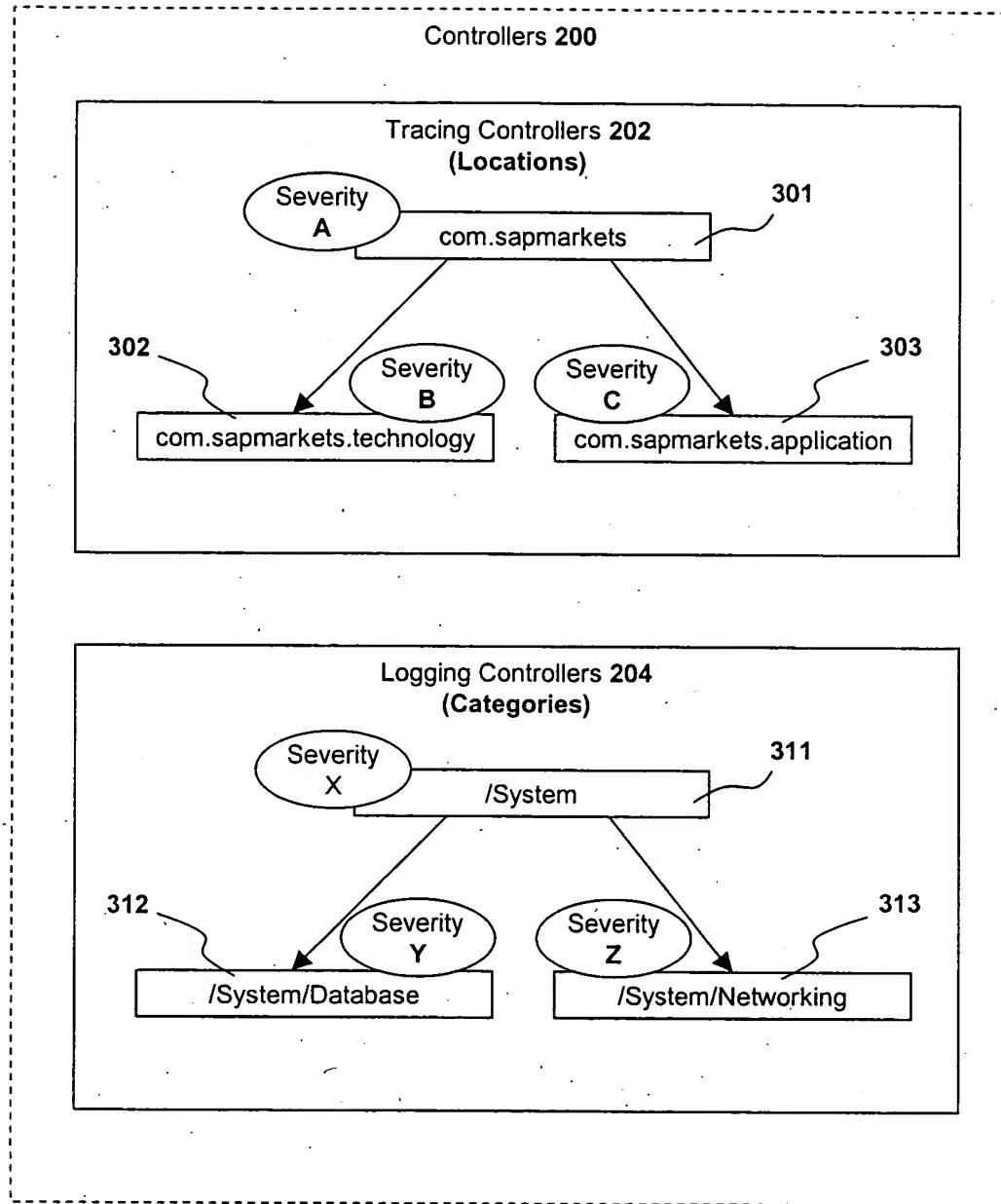


Fig. 3

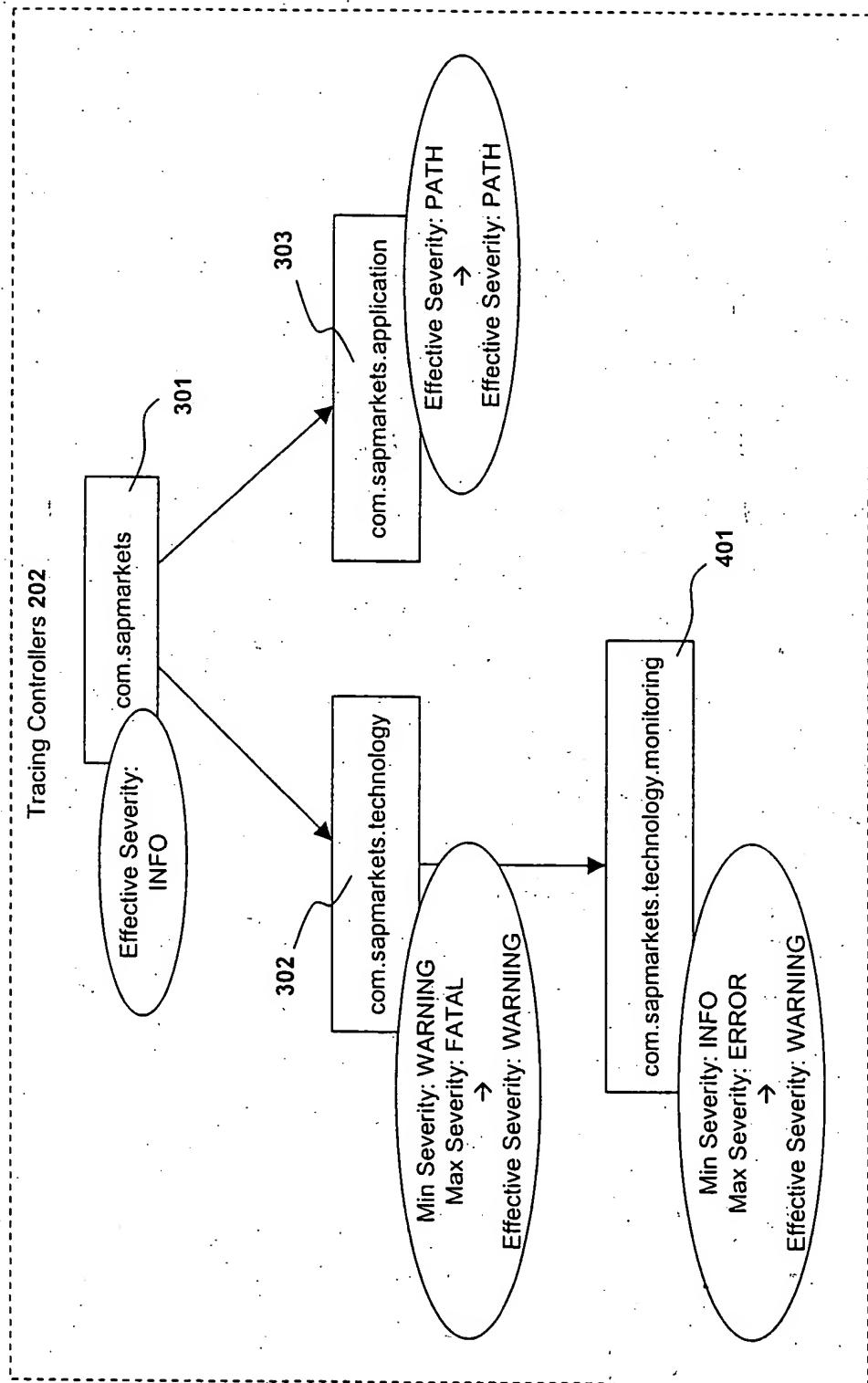


Fig. 4

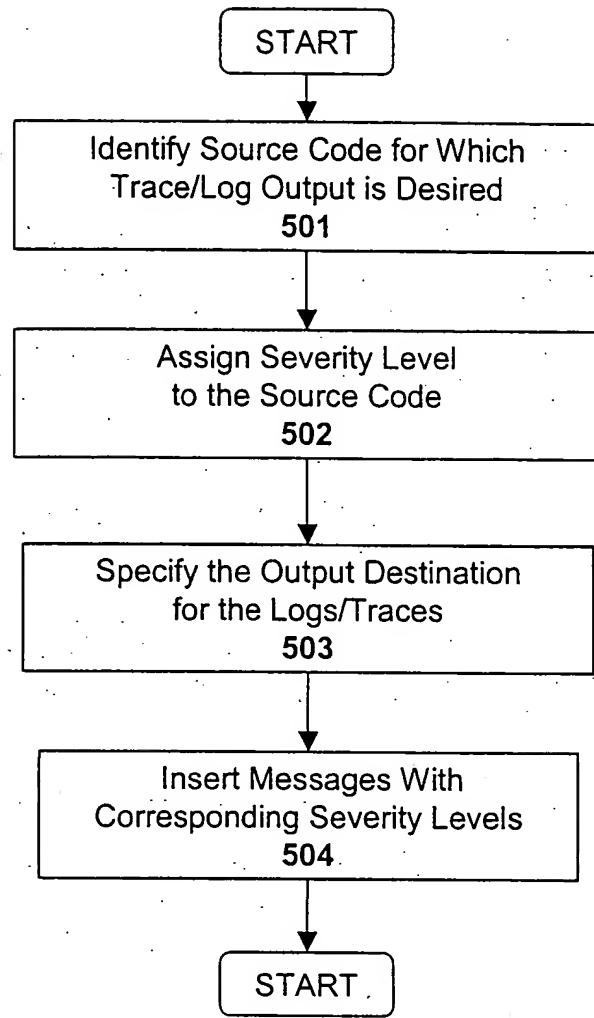


Fig. 5

MIN	0	Minimum restriction
DEBUG	100	For debugging purpose, with extensive and low level information
PATH	200	For tracing the execution flow, e.g., used in the context of entering and leaving a method, looping and branching operations
INFO	300	Informational text, mostly for echoing what has been performed
WARNING	400	Application can recover from anomaly, and fulfill the desired task, but need attention from developer/operator
ERROR	500	Application can recover from error, but cannot fulfill desired task due to the error
FATAL	600	Application cannot recover from error, and the severe situation causes fatal termination
MAX	700	Maximum restriction

Fig. 6a

ALL	0	Output messages of all severity
NONE	701	No messages will be logged

Fig. 6b

```
##### Location "com.sapmarkets.usermanagement"  
##### Assign severity threshold to WARNING level  
710 → com.sapmarkets.usermanagement.severity = WARNING 716  
##### Now, specify Console as the destination to 'usermanagement'  
720 → com.sapmarkets.usermanagement.logs = ConsoleLog 724  
720 → 700
```

FIG. 7a

```
Jan 3, 2002 6:54:18 PM com.sapmarkets.usermanagement.ClassA.method1[main] Warning: xxxxxx ...  
Jan 3, 2002 6:54:18 PM com.sapmarkets.usermanagement.ClassB.method2 [main] Fatal: Unable to ...
```

FIG. 7b

FIG. 8

```
805 → com.sapmarkets.usemanagement.severity ← 810 ← 815 ← 820 ← com.sapmarkets.usemanagement.logs = log[Console], log[File1]
```

Now, push messages simultaneously to 2 destinations:
a console and an output file
represented by a variable notation ← 825 ← 830 ← 835 →

```
log[File1].pattern = C:\Temp\myTrace.txt  
log[File1].formatter = TraceFormatter
```

Set up a FileLog, for storing trace, with <id>: 'File1'
'C:\Temp\myTrace.txt' or 'C:/temp/myTrace.txt'
Dumping messages with the default format of TraceFormatter
log[File1] = FileLog
log[File1].pattern = C:\Temp\myTrace.txt
log[File1].formatter = TraceFormatter

Set up a ConsoleLog, with <id>: 'Console'
Dumping message with self-defined format of TraceFormatter
log[Console] = ConsoleLog ← 840 ← 845 ← 850 →
log[Console].formatter = formatter[TraceNoDate]

Set up a TraceFormatter, with <id>: 'TraceNoDate'
and its pattern starts with the Severity level, and consists no date/timestamp
formatter[TraceNoDate] = TraceFormatter ← 860 ← 861 ← 862 ← 863 ← 864 ← 865 →
formatter[TraceNoDate].pattern = %s: %-30l [%t]: %m

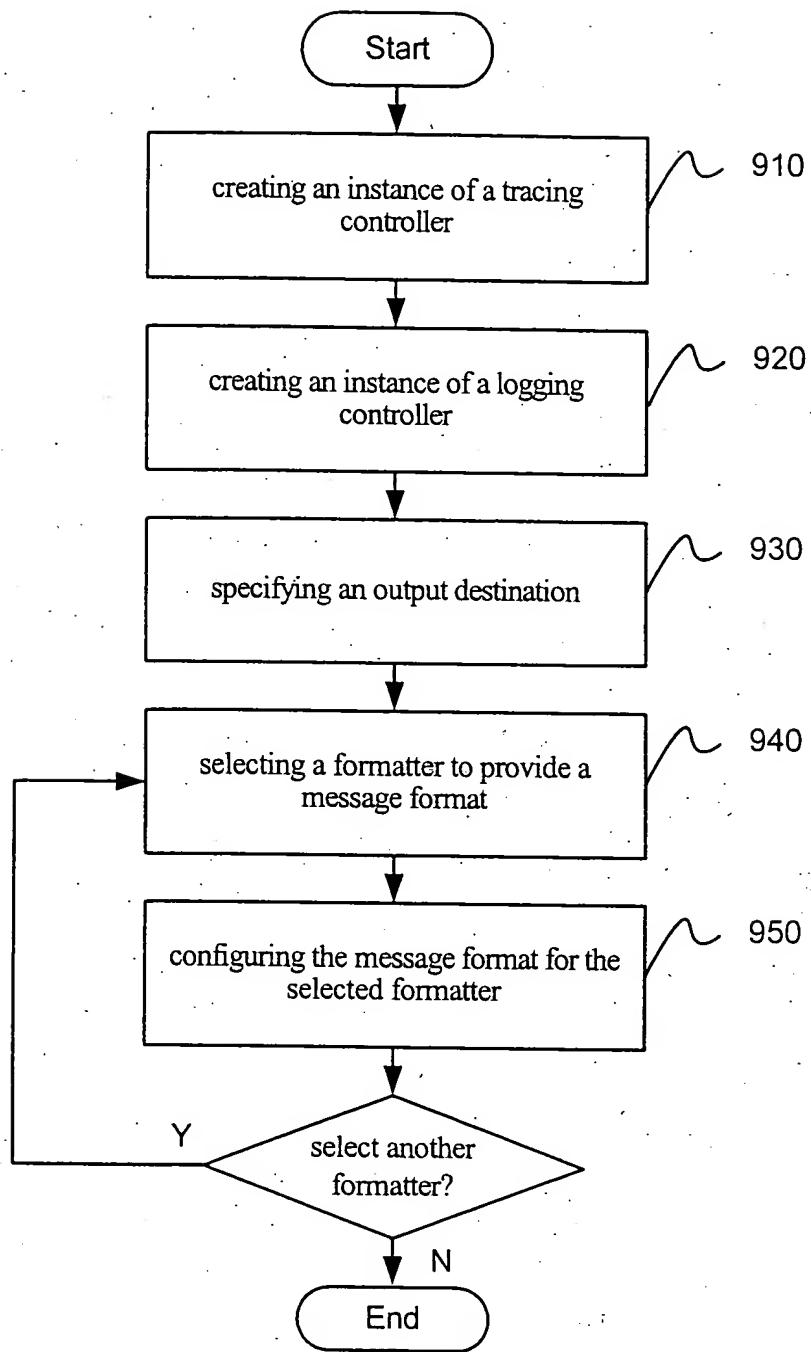


FIG. 9

FIG. 10

